On Opportunities for Mobile Offloading

Radu-Ioan Ciobanu
Faculty of Automatic Control and Computers
University Politehnica of Bucharest
Bucharest, Romania
Email: radu.ciobanu@cs.pub.ro

Ciprian Dobre
Faculty of Automatic Control and Computers
University Politehnica of Bucharest
Bucharest, Romania
Email: ciprian.dobre@cs.pub.ro

***Abstract*—Since the number of Wi-Fi-capable devices existing nowadays has increased drastically in the last few years, traffic offloading between mobile broadband and Wi-Fi has been gaining steady traction. When performed correctly, offloading should provide users with a seamless experience when they use applications on their devices, while optimizing resources to improve user experience. For this reason, in order to analyze the situations when offloading can be performed and how it should best be done, in this paper we perform an in-depth analysis of the behavior of devices with mobile broadband interfaces in various conditions, such as mobility, different times of day, varying types of traffic (audio and video streaming, HTTP). Using this information as input, we propose a device-to-device offloading method and show that using other devices as an offloading tool brings improvements in terms of processing time and server usage.**

I. Introduction

Global mobile traffic is very likely to continue growing in the years to come, thanks to the increasing popularity of smart mobile devices and the introduction of affordable data plans by cellular operators. As today's most common data access method for mobility, cellular networks are under pressure trying to cope with this data overload, coming mostly from audio, video and HTTP traffic on smartphones. Accommodating this growth requires investments in the radio access network and core infrastructure, which are not economically feasible to implement. Thus, there is a growing interest towards alternative methods to mitigate the pressure on the cellular network, and mobile data offloading proves an attractive one. Besides the obvious benefit of relieving the mobile broadband (MBB) network load, shifting data to a complementary wireless technology leads to several improvements, including increasing the overall throughput, the reduction of content delivery time, the extension of network coverage, or the increase of availability.

When offloading data, the goal is not only to reduce congestion on the broadband channel, but also to improve the performance of the communication. Thus, the media selected for offloading should offer a quality of experience (QoE) at least as good as what would have been obtained without offloading, preferably without incurring additional costs. If offloading is done on Wi-Fi, the access point (AP) employed should be carefully selected in order to lead to the expected performance, which becomes more complicated in mobility conditions, where a node only stays in range of an AP for a short period of time. An alternative method is to offload to nearby devices in an opportunistic fashion, through close-range protocols such as Bluetooth or Wi-Fi Direct. However, traffic requirements are also important, since not all traffic might be suited for offloading this way, so it is important to correctly select the offloading interface.

An open research issue is the connection between network conditions and mobility aspects. Considering the metrics that a smartphone can detect by monitoring conditions, how can it differentiate between low and high mobility? In addition, how can we differentiate between situations when a device has a sufficient density of neighbors to offload to? How can we properly estimate if the node is at the edge of the Wi-Fi cell, and mobility helps or prohibits the offloading? This paper takes the first step in attempting to answer these questions, through two main contributions. Firstly, we perform a thorough analysis of the behavior of three mobile broadband operators on static and mobile devices under various types of traffic and conditions, in order to assess the requirements for each situation. Secondly, using this data as input, we propose an offloading solution and show that it can bring benefits (in terms of computation duration and server usage) in a scenario where users generate tasks that need computing.

The remainder of this paper is structured as follows. Section II presents various approaches for offloading in mobile networks. In Sect. III, we analyze MBB data collected in different conditions, showing how several metrics vary under diverse traffic. In Sect. IV, we propose an offloading mechanism and assess its behavior on the data previously collected. Finally, Sect. V presents our conclusions.

II. Related Work

There are two general approaches to traffic offloading in cellular networks [1], [2]. Firstly, diverting traffic, data and computations through fixed Wi-Fi access points is a conventional solution to reduce traffic, as hotspots generally provide better connection speed and throughput than cellular networks [3], [4], [5], [6]. However, coverage is limited and mobility is in general constrained within the Wi-Fi cell. The increasing popularity of smart devices proposing several alternative communication options makes it possible to deploy a terminal-to-terminal (T2T) network that relies on direct communication between mobile users in an opportunistic fashion [7]. Benefiting from shared interests among co-located users, a cellular provider may decide to send popular content only to a small subset of users via the cellular network, and let these users spread the information through opportunistic contacts [8], [9], [10]. In our case, we consider the research of solutions where these two forms of offloading (AP and T2T-based) are used concurrently, enabling users to retrieve data and perform computations in a hybrid way. Advanced architectures such as MADNet [5] integrate cellular, Wi-Fi APs, and T2T communication, where the cellular network is employed as a control channel.

There are other solutions trying to combine MBB with T2T offloading in various ways. The throughput experienced by mobile users can be increased by taking advantage of neighbors with better cellular connectivity, employed as proxies. For example, users experiencing low cellular downlink channel rates can connect via ad hoc links to neighbors with better channel conditions [11]. Some architectures exploit hybrid delivery options in order to extend the range of fixed APs [12].

Based on current existing solutions, we observe a need to analyze the relationship between human mobility and the networking conditions, to come up with truly hybrid solutions where each mobile node monitors context metrics, and where the cellular network is used as a primary control mechanism. At the same time, nodes continuously exchange information between themselves to update their local knowledge of the network and device capabilities, all for the benefit of advanced prediction mechanisms capable to forecast how the network conditions will evolve. It is not only what the smartphone is able to sense and monitor that matters, but also what the network has to offer, what it can support back to the user, and how the user moves. This paper is the first step towards a real-world analysis of these specific issues.

III. Mobile Broadband Behavior Analysis

In order to make optimized decisions (when to offload, who to offload to, what to offload), it is necessary to understand the behavior of mobile networks in different conditions (at various times in a day, under mobility, when there is a high density of nodes per cell, etc.). Thus, we present a data collection experiment performed using the MONROE platform [13], [14], whose results we use as input for our offloading solution in Sect. IV. We perform a thorough analysis of the results for three MBB operators under various conditions.

A. The MONROE Platform

MONROE [13], [14] is an open access hardware-based platform for large-scale experiments in mobile broadband scenarios. It offers a large set of static and mobile nodes that have three MBB interfaces each, located in several countries. The platform allows users to schedule Linux-based containerized experiments through a Web interface at any time and repeat them when needed. Using this platform, we were able to easily collect information about MBB behavior in various conditions and then analyze it, as shown in the remainder of this section.

B. Data Collection

There are several metrics that we collect, for both static and mobile nodes, in order to view the effects that mobility and time of day have on the behavior of a mobile broadband connection. We focus on the most common activities performed using a smartphone: surfing the Internet (or using a social app), streaming music, and watching videos[1]. We have chosen Sweden as the analyzed location, because there are both static and mobile (mounted on buses that go from Karlstad to neighboring cities) MONROE nodes available.

We performed a broad set of experiments for three operators (Telenor, Telia and Tre (3)[2]), as follows. The download speed is measured using *wget*; the average round-trip time (RTT) is measured through *ping* tests; the latency and throughput of audio streaming with *mplayer* (version 2.0-728) are measured using *tcpdump* and *ifstat* (our data collector streams from an Internet radio with a 128 kbit/s rate); the latency and throughput of video streaming with *mpv* are also measured (the data collector streams a live YouTube video, in order to have constant buffering); finally, the latency and throughput of HTTP are measured using *http-ping*.

In addition to the metrics shown above, we also analyze RSRP (reference signal received power) and RSRQ (reference signal received quality), which are implicitly collected by the MONROE nodes. All these experiments were repeated hourly, for multiple days of the week. Since the mobile nodes were not always available (because the buses do not run continuously), we chose an interval when the data was the most complete, thus ending up with 7 sets of data for each hour in the 2 PM - 8 PM interval (Sweden time). We only collected data on weekdays and when the buses had the same route (for mobile nodes), in order to have similar conditions.

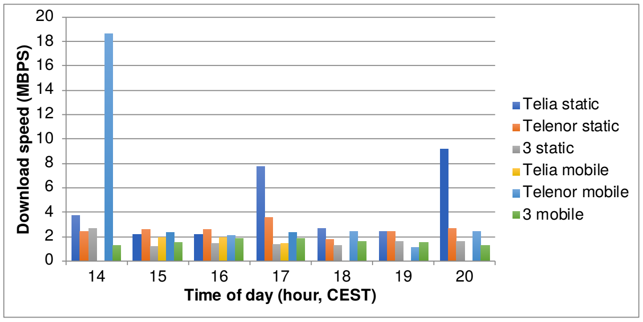
C. Data Analysis

Figure 1(a) shows the average download speed for nodes with three different Swedish operators. It can be observed that, for static nodes, Telia generally has the highest download speeds, whereas Tre has the lowest. The static node download speeds peak at 5 PM and 8 PM, which are the time periods just before and after people go home from work. For mobile nodes, Telenor tends to behave the best, reaching speeds of 18 MB/s for certain scenarios. When comparing static to mobile node performance, it can be observed that static nodes generally behave better, but the speed of Tre static nodes is mostly worse than what is obtained by mobile Telenor and Telia nodes.

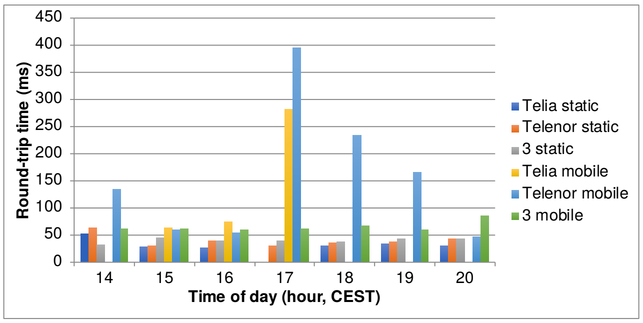
Next, Fig. 1(b) shows the average round-trip times, and it can be seen that mobile nodes (especially ones with Telenor) sometimes have very high values. Furthermore, the RTT is always higher for mobile than for static nodes, since they move between cells often. It is also interesting to observe that the mobile nodes tend to have higher RTTs in the evening (from

[1] https://bit.ly/2qXgA98

[2] https://www.telenor.se, https://www.telia.se, https://www.tre.se

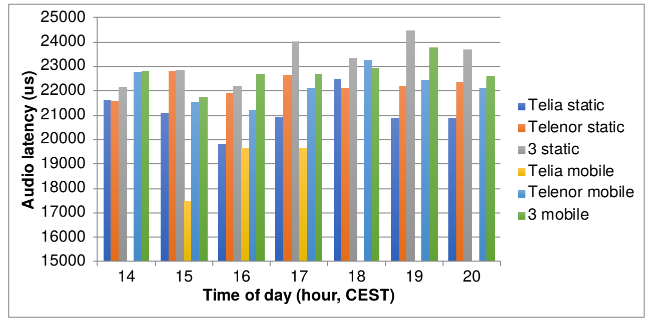


(a) Download speed

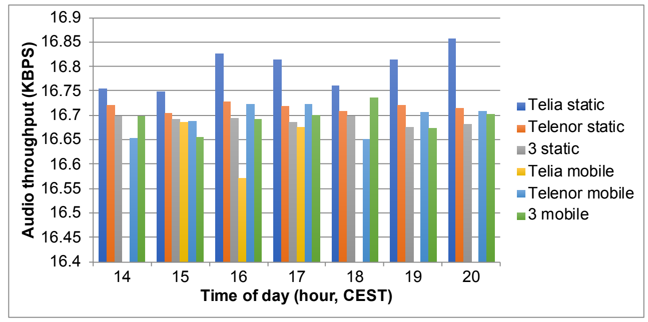


(b) Round-trip time

Fig. 1. Speed and round-trip time analysis.



(a) Latency



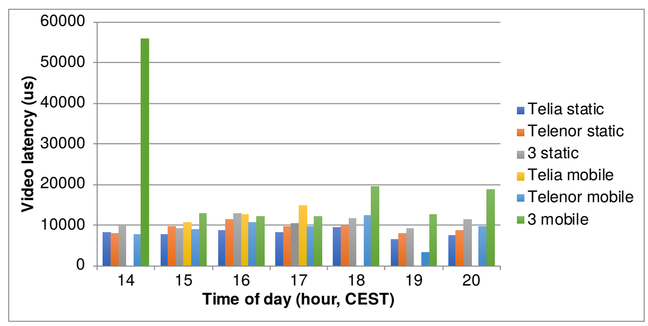
(b) Throughput

Fig. 2. Audio streaming analysis.

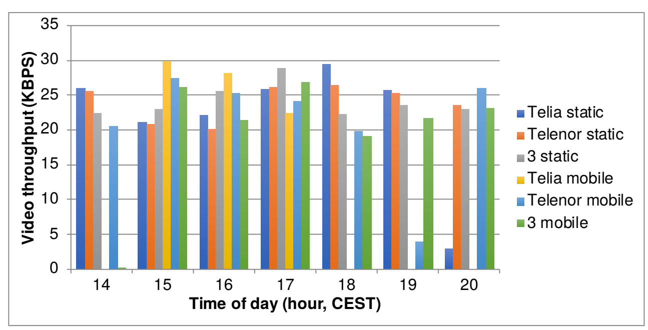
5 PM), which is generally when people start going home from work. Thus, it is highly likely that the buses are more crowded at this time than during the day, which is the cause of the spikes. For static nodes, the situation tends to be the opposite: the RTTs have higher values earlier in the day, when more people are around. Similar to the download speed analysis, Telia also has the best values for RTT for static nodes.

Figure 2 presents the behavior of mobile and static nodes for a scenario where the user performs audio streaming. It can be observed that, in terms of latency, Telia has the best values for both static and mobile devices, whereas Tre performs the worst. Another observation that can be extracted from Fig. 2(a) is that the latency values for mobile devices tend to be slightly better than the ones obtained by static devices. The audio throughput results, showed in Fig. 2(b), are also very similar between mobile and static devices and between operators, indicating that audio streaming at 128 kbit/s will not be affected too much by mobility or by less efficient operators. However, it can still be observed that static nodes behave slightly better, and also that Telia again has the best throughput for static nodes, while Tre has the worst one. For mobile nodes, Telia has the lowest throughput, while Telenor tends to behave the best.

The analysis of video streaming is presented in Fig. 3, where it can be seen that mobile nodes have higher latencies than static nodes. Again, Tre has the worst behavior for both types of nodes, whereas Telia obtains the best values for static nodes and Telenor for mobile nodes. It is also interesting to note that the best latency values are obtained in the evening (at 7 PM and 8 PM). Another conclusion that can be drawn



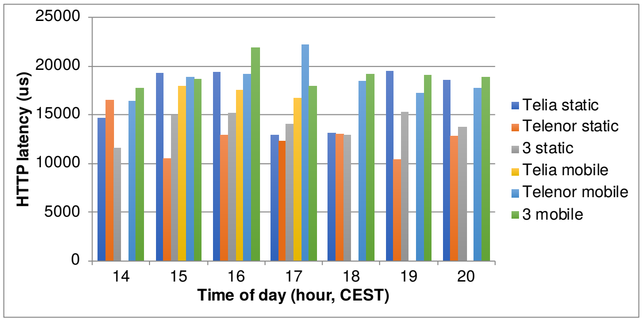
(a) Latency



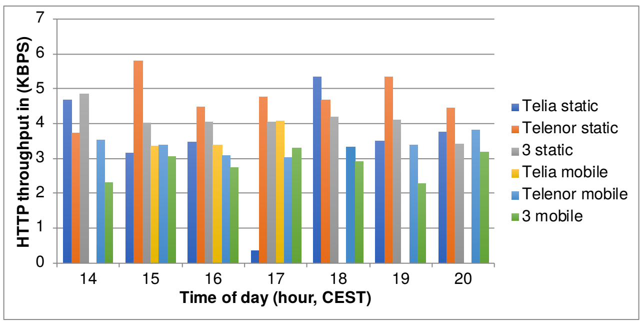
(b) Throughput

Fig. 3. Video streaming analysis.

when comparing Fig. 3(a) to Fig. 2(a) is that video streaming yields lower latencies than audio streaming for most of the cases. Figure 3(b) shows that mobile nodes exhibit better video throughput than static nodes at earlier hours (i.e., between 3 PM and 5 PM), while later it can be seen that static nodes

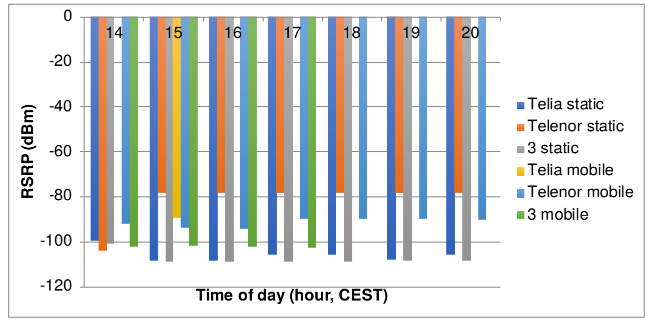


(a) Latency

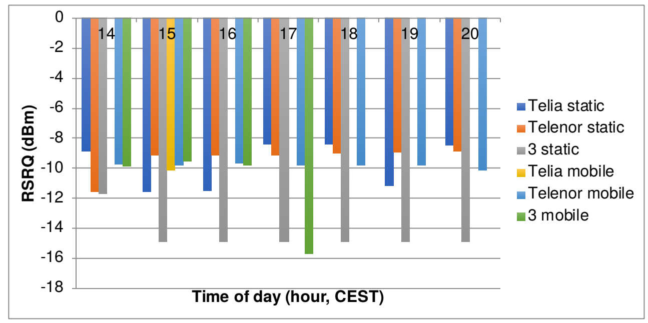


(b) Throughput

Fig. 4. HTTP analysis.



(a) RSRP



(b) RSRQ

Fig. 5. RSRP and RSRQ analysis.

behave better, with Telia having the highest values.

The HTTP scenario is shown in Fig. 4. It can be seen that, for static nodes, Telenor has the lowest latencies, whereas Telia has the worst values. It should also be noted that high static latencies tend to show up in the evening (7 PM and 8 PM) or in the early afternoon (3 PM and 4 PM). For mobile nodes, Telia generally has the best latency values. Regarding HTTP throughput, as shown in Fig. 4(b), static nodes (especially Telenor) have clearly better values than mobile nodes.

Finally, we look at two relevant metrics for measuring a 4G signal, namely RSRP (reference signal received power) and RSRQ (reference signal received quality). The former is defined as the average of power contributions of the resource elements that carry cell-specific reference signals within the considered measurement frequency bandwidth[3], and is generally used to rank various candidate cells based on their signal strength. A 4G connection is considered excellent when the RSRP values are above -80 dBm. The RSRQ indicates the quality of the received reference signal, providing additional information when RSRP is not enough to make a cell handover decision. When the RSRQ is above -10 dBm, the quality of the signal is considered excellent.

In Fig. 5(a) it can be noticed that only Telenor on static nodes has an RSRP value above the threshold for an excellent connection. Other static values are around -100 dBm, which means that the signal power is weak and that there are many disconnections. However, we have previously seen that the best values for several metrics (such as download speed, RTT, audio/video latency and throughput) for static nodes have been obtained by Telia. This shows that using the signal metrics is not enough to obtain a good view of the QoE, but that other metrics are required for a complete picture. Figure 5(a) also shows that Tre has the lowest values for RSRP on static nodes, which correlates with the fact that this operator has exhibited the worst values in terms of download speed, audio latency and throughput, video latency, and HTTP latency. This means that Tre does not have a good coverage in the locations of the static MONROE nodes. However, it can be observed that Tre exhibits better RSRP values for mobile nodes than for static nodes, but this does not necessarily translate into a better performance for mobile nodes, because the fact remains that cell handover has to be performed often in mobility conditions.

In terms of RSRQ, as seen in Fig. 5(b), most of the mobile nodes show an excellent signal quality. In the majority of cases, the mobile RSRQ is better than the static RSRQ, the exception being Telenor, which again has the best values for static nodes. On the other hand, Tre tends to have very high RSRQ values for static nodes, showing a poor signal quality.

D. Conclusions

The analysis presented in Sect. III-C shows that, for various scenarios (audio streaming, video streaming, HTTP), the behavior of the three operators studied varies based on time of day, the mobility of nodes, and other such factors, making offloading a necessity in some cases. As discussed above, for static nodes Telia seems to behave the best in terms of download speed, RTT, audio/video latency and throughout, whereas Telenor has the best results for HTTP latency and

[3]https://bit.ly/2F0APG3

throughput. For mobile nodes, Telenor is best in terms of download speed, audio throughput, video latency, while Telia has the top results for audio latency, video throughput, HTTP latency. We also observed that the RSRP and RSRQ are not entirely correlated with other metrics such as download speed, RTT, throughput or latency, and that there are other factors that affect the behavior and perceived QoE of an MBB interface.

Another important observation that we made when analyzing the data is that these results tend to hold for the same intervals in different weekdays. For example, although the download speed for a certain day at 5 PM differs from the same metric measured one hour later, a similar value will be obtained one day later (i.e., in the same weekday) at 5 PM. This shows that, at least based on our data collected so far, there is a certain predictability in the behavior of MBB interfaces, and this is a topic that we wish to study in depth in the future. Such a predictability would help in the offloading process, because the mechanism would be able to learn from past behavior and make the offloading decisions in advance.

IV. OFFLOADING

In this section, we want to show the benefits of data offloading in mobile networks where the conditions can change (due to mobility and other factors) as shown in Sect. III. For this reason, we propose a basic offloading scheme, where mobile nodes have tasks that they need to compute, and which they can either process themselves, upload to a server or a cloud and wait for the result, or leverage other neighboring nodes in an opportunistic T2T fashion.

A. *Offloading Scheme*

The offloading scheme we propose works as follows. When a node is idle, it generates some tasks, which are processing items of various sizes (such as applying a filter to a photo, computing the next move in a game of chess, etc.). Normally, the default way mobile apps approach such a situation is by uploading the tasks to the cloud and waiting for the results. However, as we have seen in Sect. III, there are some situations where a node's connection is not particularly good, and where other approaches might be more suitable. Thus, in this paper, we propose an offloading scheme where nodes attempt to leverage neighboring devices instead of going directly to the cloud. The main advantage would be that the latency of using close-range protocols can be lower than that of an MBB interface, so the computations would be performed faster through the help of other nodes. Furthermore, a neighboring node might already have the computation result or information required, which would further increase the processing speed. Additional benefits include lowering the data plan consumption (since close-range protocols are used instead of 4G) and the cloud usage (benefiting application developers).

Thus, our offloading scheme assumes that nodes do not send their tasks directly to the cloud, but instead wait for a certain period of time to see if any other node that comes in proximity can help with the computations. Whenever two nodes come in range, they check to see if the tasks that they need to compute are balanced (i.e., if the total estimated durations of their computations are roughly equal). If there is no balance, then they exchange tasks in order to achieve a fair distribution. When a node finalizes a task, it attempts to get its result back to the task's owner until the time limit expires. If a node has sent its task to another node for computation but the time limit expires, it computes the task in the cloud.

B. *Testing Scenario*

Using the MobEmu simulator[4], we tested our offloading scheme on two separate mobility scenarios, with the collected data presented in Sect. III as input. The first scenario uses the HCMM mobility model [15] to simulate the interactions between 36 mobile nodes grouped into 4 communities for a duration of 7 hours (corresponding to the 2 PM - 8 PM interval of hourly datasets collected from the MONROE nodes). We simulated a 400x400-meter area, with a node's speed between 1.25 and 1.5 m/s and a proximity range of 10 meters, leading to a total of 16134 contacts between devices. We considered that the first twelve nodes have Telia, the next twelve have Telenor, and the final twelve have Tre, mapping the data collected from the mobile MONROE nodes accordingly.

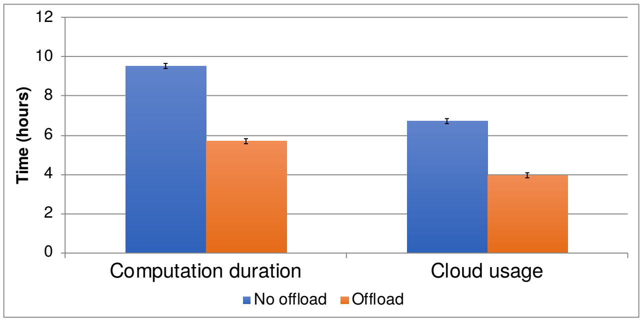
For the second scenario, we used a real-life mobility trace collected in an office environment in Bucharest in 2015 using the HYCCUPS Tracer application[5]. We selected a sample from this trace which contains contacts collected from 6 nodes during a 7-hour window in a workday, and we mapped the data collected from the mobile MONROE nodes as input (each operator analyzed in Sect. III is present on two nodes). There were 324 total contacts between the nodes.

We considered that nodes generate three types of tasks (of 1, 1000 and 10000 Mcycles, respectively), that they use one 2.3 GHz core for computations, and that the cloud has an unlimited number of single-core 3.3 GHz virtual machines available. The time limit for uploading tasks to the cloud is one millisecond for small tasks, one second for medium tasks, and ten seconds for large tasks). We also set the size of a task to 5 MB and the T2T transfer speed to 3 MB/s.

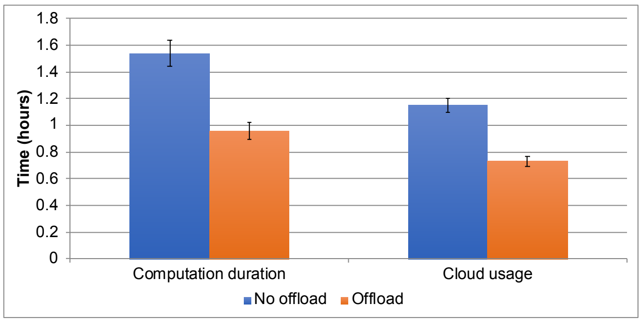
C. *Results*

The results for the HCMM simulation are shown in Fig. 6(a). We measured the total computation duration (i.e., the sum of times between the moment a task is generated by a node and the moment the node receives its result, either from the cloud or from another node) and the time spent computing tasks in the cloud, which would translate into costs and energy consumption for the application developers. Fig. 6(a) thus shows not only that the total computation duration decreases by 3.7 hours (i.e., 40%) when mobile nodes offload data, but also that the cloud usage is reduced by 2.7 hours (i.e., 41%). This means that a node's tasks are solved faster when employing neighboring devices, so the user experience is increased. Furthermore, the data plan consumption will be lowered, since close-range protocols such as Bluetooth or

[4]https://github.com/raduciobanu/mobemu
[5]http://www.smartrdi.net/2017/11/08/getting-started/



(a) HCMM



(b) Office trace

Fig. 6. Offloading results.

Wi-Fi Direct are used instead of 4G. The app developers also benefit from this, since their cloud provider bill will be lower if the cloud usage is reduced. Our experiments also showed that the battery consumption is reduced by 10% when devices offload their tasks.

The experiments performed on the office trace, presented in Fig. 6(b), show similar results. Namely, it can be seen that offloading reduces the total computation time by 37% and the cloud usage time by 36%. This shows that offloading works in various conditions, since the two scenarios differ in terms of density and node behavior.

V. Conclusion

In this paper, we performed a detailed analysis regarding the behavior of three mobile broadband operators in Sweden in various mobility and traffic scenarios. Based on this information, we proposed a data and computation offloading method that uses neighboring devices as relays whenever possible. The results obtained showed that, for two different scenarios based on the data analyzed at the first stage, performing offloading improves the computation duration (i.e., the overall user experience) and decreases the server usage time (thus lowering the infrastructure costs of mobile app developers, while at the same time reducing their carbon footprint).

Acknowledgment

This work is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 644399 (MONROE) through the open call project "Traffic and Data Offloading in Mobile Networks: TTOff". The views expressed are solely those of the authors. This research is also supported by project SPERO (PN-III-P2-2.1-SOL-2016-03-0046, 3Sol/2017) and by University Politehnica of Bucharest, through the "Excellence Research Grants" program, UPB - GEX 2017, identifier UPB-GEX2017, ctr. no. AU 11.17.02/2017.

References

[1] F. Rebecchi, M. D. de Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, "Data offloading techniques in cellular networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 580–603, Secondquarter 2015.

[2] A. Aijaz, H. Aghvami, and M. Amani, "A survey on mobile data offloading: technical and business perspectives," *IEEE Wireless Communications*, vol. 20, no. 2, pp. 104–112, April 2013.

[3] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: How much can wifi deliver?" *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 536–550, April 2013.

[4] X. Kang, Y.-K. Chia, S. Sun, and H. F. Chong, "Mobile data offloading through a third-party wifi access point: An operator's perspective," *IEEE Transactions on Wireless Communications*, vol. 13, no. 10, pp. 5340–5351, Oct 2014.

[5] S. Dimatteo, P. Hui, B. Han, and V. O. Li, "Cellular traffic offloading through wifi networks," in *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, Oct 2011, pp. 192–201.

[6] P. Raveneau, R. Stanica, M. Fiore, S. Uppoor, M. Cunche, H. Rivano, and Z. Smoreda, "Urban-scale cellular offloading through wi-fi access points: A measurement-based case study," in *2015 IEEE 1st International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, Sept 2015, pp. 132–137.

[7] R.-C. Marin, R.-I. Ciobanu, C. Dobre, C. X. Mavromoustakis, and G. Mastorakis, "A context-aware collaborative model for smartphone energy efficiency over 5g wireless networks," *Computer Networks*, vol. 129, pp. 352 – 362, 2017, special Issue on 5G Wireless Networks for IoT and Body Sensors.

[8] F. Mehmeti and T. Spyropoulos, "Is it worth to be patient? analysis and optimization of delayed mobile data offloading," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, April 2014, pp. 2364–2372.

[9] S. Eido and A. Gravey, "How much lte traffic can be offloaded?" in *Meeting of the European Network of Universities and Companies in Information and Commun. Engineering*. Springer, 2014, pp. 48–58.

[10] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, "Mobile data offloading through opportunistic communications and social participation," *IEEE Transactions on Mobile Computing*, vol. 11, no. 5, pp. 821–834, May 2012.

[11] H. Luo, X. Meng, R. Ramjee, P. Sinha, and L. Li, "The design and evaluation of unified cellular and ad-hoc networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 9, pp. 1060–1074, Sept 2007.

[12] M. Pitkanen, T. Karkkainen, and J. Ott, "Opportunistic web access via wlan hotspots," in *2010 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, March 2010, pp. 20–30.

[13] Ö. Alay, A. Lutu, R. García, M. Peón-Quirós, V. Mancuso, T. Hirsch, T. Dely, J. Werme, K. Evensen, A. Hansen, S. Alfredsson, J. Karlsson, A. Brunstrom, A. S. Khatouni, M. Mellia, M. A. Marsan, R. Monno, and H. Lonsethagen, "Measuring and assessing mobile broadband networks with monroe," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 IEEE 17th International Symposium on A*. IEEE, 2016, pp. 1–3.

[14] Ö. Alay, A. Lutu, M. Peón-Quirós, V. Mancuso, T. Hirsch, K. Evensen, A. Hansen, S. Alfredsson, J. Karlsson, A. Brunstrom, A. Safari Khatouni, M. Mellia, and M. Ajmone Marsan, "Experience: An open platform for experimentation with commercial mobile broadband networks," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 2017, pp. 70–78.

[15] C. Boldrini and A. Passarella, "Hcmm: Modelling spatial and temporal properties of human mobility driven by users' social relationships," *Comput. Commun.*, vol. 33, no. 9, pp. 1056–1074, Jun. 2010.